Optical map guided genome assembly

Leinonen, Miika

2020-07-06

Leinonen , M & Salmela , L 2020 , ' Optical map guided genome assembly ' , BMC Bioinformatics , vol. 21 , 285 . https://doi.org/10.1186/s12859-020-03623-1

http://hdl.handle.net/10138/321132
https://doi.org/10.1186/s12859-020-03623-1

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

Optical map guided genome assembly

Miika Leinonen and Leena Salmela*

*Correspondence:
leena.salmela@cs.helsinki.fi
Department of Computer Science, Helsinki Institute for Information Technology, University of Helsinki, Pietari Kalmin katu 5, Helsinki, Finland

Abstract

Background: The long reads produced by third generation sequencing technologies have significantly boosted the results of genome assembly but still, genome-wide assemblies solely based on read data cannot be produced. Thus, for example, optical mapping data has been used to further improve genome assemblies but it has mostly been applied in a post-processing stage after contig assembly.

Results: We propose OPTICALKERMIT which directly integrates genome wide optical maps into contig assembly. We show how genome wide optical maps can be used to localize reads on the genome and then we adapt the Kermit method, which originally incorporated genetic linkage maps to the miniasm assembler, to use this information in contig assembly. Our experimental results show that incorporating genome wide optical maps to the contig assembly of miniasm increases NGA50 while the number of misassemblies decreases or stays the same. Furthermore, when compared to the Canu assembler, OPTICALKERMIT produces an assembly with almost three times higher NGA50 with a lower number of misassemblies on real *A. thaliana* reads.

Conclusions: OPTICALKERMIT successfully incorporates optical mapping data directly to contig assembly of eukaryotic genomes. Our results show that this is a promising approach to improve the contiguity of genome assemblies.

Keywords: Genome assembly, Optical mapping

Background

The long reads produced by third generation sequencing technologies such as Pacific Biosciences and Oxford Nanopore have enabled large improvements in *de novo* genome assembly. Nevertheless, assemblies produced solely on read data are not complete and typically contain orders of magnitudes more contigs than the sequenced organism has chromosomes. To further improve these assemblies, several long-range technologies such as optical mapping, genetic linkage maps, and Hi-C based analysis have been developed [1].

Here we concentrate on using optical mapping data to improve genome assembly. Optical maps are produced by fragmenting the genome to produce hundreds of kilobases long DNA molecules. Each DNA molecule is then elongated on a plate. A restriction enzyme which cuts at a specific DNA motif is applied on the DNA molecules and the order and

length of the resulting fragments are measured by imaging [2, 3]. This results in raw optical mapping data which is then assembled to genome-wide optical maps.

Nowadays the optical mapping data is commonly utilized after contig assembly to further scaffold the contigs. We are aware of only two works attempting to use optical mapping data during contig assembly: AGORA [4] and KOOTA [5]. These tools were tested only on small genomes but the experiments showed that integrating optical mapping data to contig assembly can be beneficial.

Here we present OPTICALKERMIT, a contig assembler using both Pacific Biosciences sequencing reads and a genome-wide optical map. Similar to KOOTA [5], we first locate the reads on the genome-wide optical map. Whereas KOOTA mapped the in silico digested reads directly to the optical map, we first create preliminary contigs and use them to get more accurate location information for the reads. Finally, we assemble the reads augmented with approximate location information using the guided assembly approach developed in Kermit [6].

Our experiments show that using the genome-wide optical map increases the NGA50 of the assembled contigs as compared to assembling just the reads with the same assembly method. Furthermore, the number of misassemblies decreased or stayed the same. When compared to the Canu assembler [7] on real *A. thaliana* reads, OPTICALKERMIT produces an assembly with almost three times higher NGA50 and a lower number of misassemblies.

OPTICALKERMIT is freely available at https://github.com/Denopia/kermit-optical-maps.

Related work

An optical map is a sequence of lengths of subsequent DNA sequence fragments. A DNA sequence can be cut at specific places, *restriction sites*, using *restriction enzymes*. Applying a restriction enzyme to a DNA molecule cuts it into fragments at the corresponding restriction sites. For example, the enzyme XhoI recognizes the nucleotide sequence 'CTCGAG', and cuts the DNA molecule after the first C-nucleotide. This process leaves us with a number of consecutive DNA fragments, whose order is known, and their length can be measured. The measured lengths put together in order give us an optical map of the DNA sequence. A simplified example of a DNA sequence and its optical map can be seen in Fig. 1. The optical map of a genome we want to assemble is usually generated this way in a laboratory environment. In the case we have access to the DNA sequence itself, we can use in silico digestion. This means that the sequence is fragmented computationally by finding each occurrence of a subsequence corresponding to a restriction enzyme and cutting it at these sites. For example, optical maps for reads and contigs can be acquired this way since their sequences are known.

A basic task in processing optical mapping data is to align the optical maps against each other or to align in silico digested contigs to an optical map. Work in this area was pioneered by Valouev et al. [8] who developed a dynamic programming algorithm to solve the alignment problem. A similar approach was later used in SOMA [9]. Because of the quadratic time complexity, these approaches can be slow. Thus several methods have been developed to align optical mapping data more efficiently. OMBlast [10] uses a seed and extend approach for alignment. Maligner [11] offers two alignment modes: a sensitive

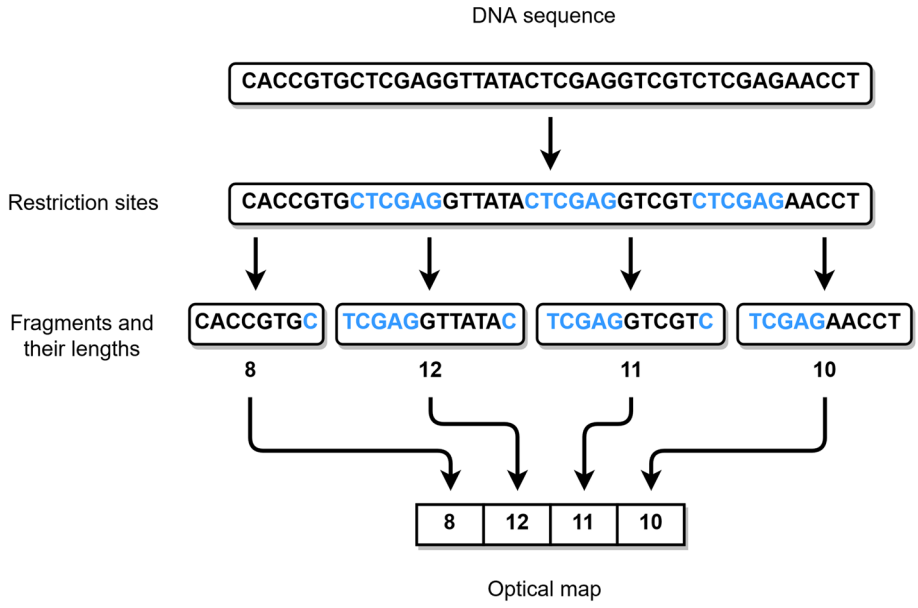

Fig. 1 Example DNA sequence and its optical map corresponding to XhoI restriction enzyme. XhoI recognizes 'CTCGAG' sites and cleaves the sequence after the first C-nucleotides. The lengths of the resulting fragments are then measured to compose the optical map

mode based on dynamic programming and an efficient indexing based approach that tolerates unmatched cutting sites in the reference but not in the query optical map. TWIN [12] uses an FM-index to facilitate efficient alignments, whereas KOHDISTA [13] indexes the optical maps as an automaton. Once in silico digested contigs have been aligned to a genome-wide optical map, the alignments can be used to detect misassemblies [14] or to order the contigs into scaffolds [15].

Optical maps have been used in several genome projects to improve the contiguity of the assembly, see e.g. [16–21]. However, optical mapping data has usually been used in a post-processing step after the contigs have been constructed. Some preliminary research has been done to involve optical maps already during the contig assembly process. For example, AGORA [4] is an assembler program that can utilize optical maps, but it was only tested with error-free reads of very small bacterial genomes. Nevertheless, AGORA got positive results for using optical maps. Another example program called KOOTA [5] also takes advantage of optical maps during assembly. While KOOTA did not perform competitively compared to the other current assemblers, it demonstrated that optical maps can be used to improve specific phases of the assembly process.

In our approach, we will use Kermit [6] to implement a guided assembly. Kermit was initially developed for genetic linkage maps. A genetic linkage map consists of a set of markers, e.g. SNVs, on a genome. Typically the markers are divided into chromosomes and within each chromosome, the markers are further divided into ordered bins. The markers of a genetic linkage map are derived from a sequenced cross which is a population of related individuals. The markers are then assigned to chromosomes and bins within chromosomes based on the observed hereditary patterns. The bins are ordered, which means that if two markers are in different bins, we know which marker appears before the other in the genome, while nothing can be said about the relative order of markers within

a single bin. Kermit assigns colors to bins, represented as integers starting from 0. If the integer, in other words, a color, of a bin is smaller than another bin's, we know that all markers within it appear before the markers in the other one. Markers in a bin are given the color of the bin they reside in. Kermit takes as input a set of sequencing reads and a genetic linkage map. It then maps the markers of the genetic linkage map to the reads and assigns reads to bins based on these mappings. During contig assembly, the assignment of reads to bins is used to produce longer contigs than would be possible solely based on the reads.

Results

Overview of our method

Our method takes as input high error rate third generation sequencing reads such as those produced by the Pacific Biosciences sequencing technology and a genome-wide optical map of the target genome. Our aim is to align the reads to the optical map and then use this location information to perform guided genome assembly. The approximate location information of reads will be expressed as colors. In our case, each fragment of the genome-wide optical map has its own color. Thus after the reads have been aligned to the genome-wide optical map, we know which fragments they overlap and we can color them with the corresponding colors. The Kermit assembler [6] can then use this color information for guided genome assembly and thus produce contigs that represent the reference genome more accurately.

To align the reads to a genome-wide optical map, two problems need to be overcome: (i) many of the cutting sites have been confounded in the reads because of the high abundance of sequencing errors and (ii) the reads are too short to unambiguously align directly to the genome-wide optical map.

To solve the first problem, we use CONSENT [22] to correct the reads. To solve the second problem, we use the corrected reads directly with a *de novo* assembler to acquire *pre-coloring contigs*, contigs that are assembled using non-colored reads. After this in silico digested optical maps of the pre-coloring contigs are created and aligned to the reference genome optical map. With optical map alignments, we are able to approximate the locations of the pre-coloring contigs within the reference genome and color them accordingly. Most of the contigs can be colored because they are much longer and their optical maps have more fragments compared to the reads. Next, the reads are aligned to the pre-coloring contigs. Alignments contain information on how the reads and contigs overlap, i.e. which contig a read aligns with and in which positions the overlap starts and ends. With this information and colored pre-coloring contigs, the reads can be colored. Afterward, the colored reads are ready to be given to the Kermit assembler as input. Kermit outputs new *post-coloring contigs*, which are the final product of our guided genome assembly, representing the reference genome more accurately than the pre-coloring contigs. The whole assembly workflow is shown in Fig. 2.

Datasets

We used three different sets of reads to test our guided genome assembly pipeline. One set contained reads of *A. thaliana* obtained by PacBio sequencing. The two others were simulated reads of *S. cerevisiae* and *C. elegans* obtained by SimLoRD [23], which is a read generation software mimicking the error pattern of PacBio sequencing.

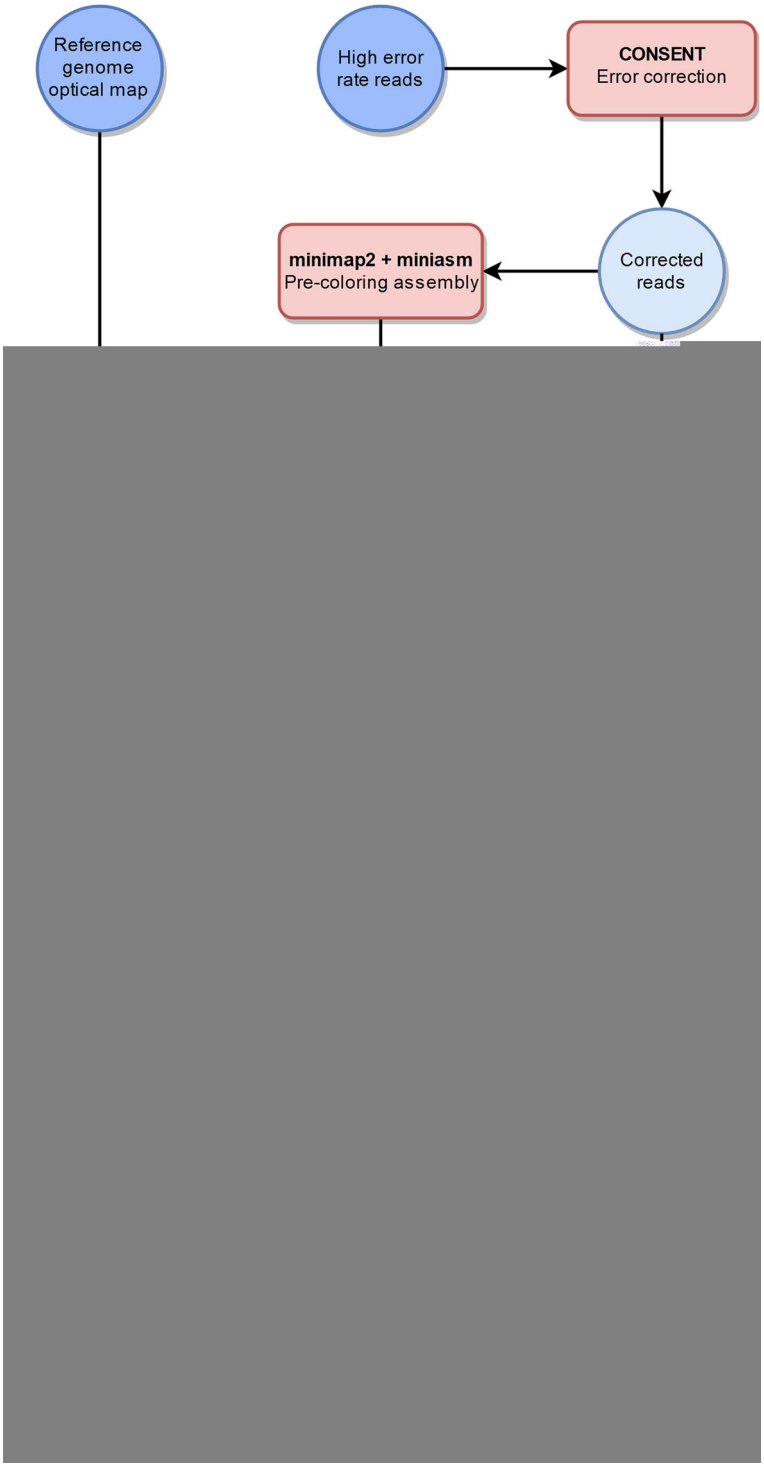

Fig. 2 OPTICALKERMIT assembly workflow

All of our read sets had a genome coverage of 40x. *A. thaliana* reads were chosen from the larger set of reads from longest to shortest until we got to the target coverage. Information about the reference genomes and data sets can be seen in Table 1.

Minimap2 can produce multiple possible alignments for a single read. We decided each read should be colored based on one alignment only, and the intuitive solution is to use the 'best' alignment for coloring. After the initial experiments seen in Table 3, we decided on a simple heuristic. A read is colored based on the alignment with the greatest number of matching bases between it and the aligning section of the contig. If the number of matching bases was less than 80% of the whole length of the read, the read was left uncolored.

For each alignment, the start and end positions in a contig are known, and they are used to determine which colors of the contig are used to color the read. For example, suppose the aligning section in the contig starts from position i and ends at position j. We begin to add the lengths of the contig fragments together one by one. The first fragment n, for which the sum of the lengths of the contig fragments $[0, n]$ is greater than i is chosen as the starting fragment i.e. the starting color. Similarly, the first fragment m, for which the sum of the lengths of the contig fragments $[0, m]$ is greater than j, is used as the end fragment i.e. the end color. As Kermit only uses the start and end colors of a read, this coloring scheme suffices.

We deliberated if the beginning and end parts of the reads that are left outside of the aligning section should be used to extend the coloring beyond the aligning section in the contig. For example, suppose the alignment in the contig starts again at position i and ends at position j. Additionally, suppose we also know that there is a sequence of length a in the beginning of the aligning section of the read, and a sequence of length b at the end of aligning section of the read. The question is, would it be beneficial to adjust the starting position of the aligning section in the contig to start at i Š a and to end at $j + b$ to possibly extend read colors. Figure 4 shows how extending could affect the coloring. As we expected, the coloring experiments revealed that this does not greatly affect the quality of the resulting post-coloring contigs. If the extensions would have a huge impact on the coloring, it would suggest that the aligning section itself would be short, which should not be possible because we are requiring it to have at least 80% matching bases. Nevertheless, we decided to include the extensions in our pipeline.

After the reads are colored, some of the available colors in the contigs can be completely unused by the reads. In other words, none of the read alignments overlapped with the

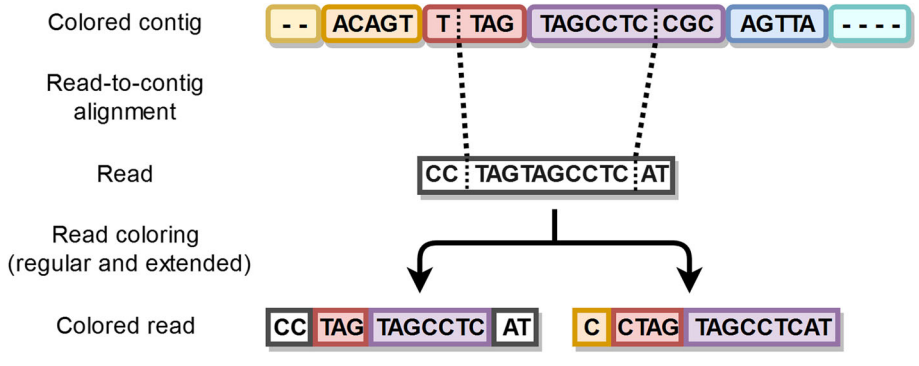

Fig. 4 Simplified example of the two different cases of read coloring when a reasonably strong alignment with a colored contig is found. **Case 1:** Left hand side coloring is based on the aligning segment only. **Case 2:** Right hand side coloring is based on the aligning segment extended with the lengths of the non-aligning sections of the read

fragments corresponding to the missing colors. To be clear, colors between the start and end colors of a read are also considered as used. It is possible that two reads are in reality physically close to each other, even though there is a color gap between them due to inaccurate alignments for example. This might affect the performance of Kermit negatively, so we decided to shift the colors so that there were no missing colors.

As a simplified example, suppose all the reads used colors $\{1, 2, 4, 7, 8, 10\}$, while the colors appearing in the reference were $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. These read colors would then be mapped to consecutive colors $\{1, 2, 3, 4, 5, 6\}$ in their respective order; 1 1, 2 2, 4 3, 7 4, 8 5, 10 6. Then the already colored reads would be recolored according to these mappings. They would still keep their respective order, but the gaps caused by unused colors get removed. Colors were only shifted within a single chromosome's colors i.e. the $s \cdot k$ term in the colors was left untouched. We ran experiments with and without this adjustment to determine its effects, and the results suggest that readjusting the colors is beneficial.

Post-coloring contig assembly

After all the data is gathered and processed to get the corrected and colored reads, we can start assembling post-coloring contigs. Our OPTICALKERMIT assembly pipeline uses the Kermit assembler [6] to do this. Kermit works very similarly to the miniasm assembler [26], the difference being that during the layout step Kermit cleans the assembly graph based on the given colorings.

Kermit starts by building an overlap graph with the help of minimap2 [24] alignments. The same read-to-read alignments that were used during the pre-coloring assembly can be used here. Each vertex of the graph represents a DNA sequence, and we know which read is responsible for it. Since the reads have been colored, the vertices can also be colored accordingly, leading to a colored overlap graph.

As with miniasm, a unitig is again defined as a maximal non-branching path in the overlap graph. However, with the vertex color information, we can alter the paths that are used to build the unitig. An edge from vertex v_i to vertex v_{i+1} means the corresponding sequences overlap, and can be merged together to be used as a part of a unitig. Some of the connections are bound to be erroneous, which can be detected with the help of the colors.

Suppose an edge (v_i, v_{i+1}) exists in the graph, but the color number of vertex v_i was larger than that of vertex v_{i+1}. The color information suggests that v_i should appear after v_{i+1}, but the edge says they are connected together in the reverse order. We trust the coloring information acquired from the optical maps more than the overlap edge, and it is removed completely from the graph. Even if the order of the colors is correct, their distance can cause suspicion. If the colors are very far apart, we can deduce the vertices should not be connected with an edge, and such edges are also removed from the graph. These kind of contradicting edges are called *inconsistent* edges.

A *consistent* edge (v_i, v_{i+1}) is an edge such that at least one of the colors of vertex v_{i+1} is equal to or exactly one greater than at least one of the colors in vertex v_i. By default Kermit only allows the colors to differ by at most one which is the option we used, but this restriction can be adjusted by the user. All edges that do not follow this requirement are discarded. This way we cannot take a path with a huge gap or where some of the sequences are in the wrong order.

Some of the reads might be left uncolored due to difficulties and uncertainties during the coloring step. An edge that is connected to an uncolored vertex would be automatically removed, but Kermit alleviates this problem of uncolored vertices by allowing the colors to propagate. An uncolored vertex is assigned all colors of the vertices that are reachable from it by traveling only through uncolored vertices. There is a limit on how far the colors should propagate, since allowing the colors to flow as far as possible would most likely lead to incorrect colorings. By default Kermit allows the colors to propagate through five vertices, but this can again be changed to the user's liking. If propagated colors have some missing colors between them, the vertex is deleted completely from the graph because the propagated coloring is not coherent which makes us suspicious of its correctness. We experimented with the default option and the no propagation option, and the results can be seen in Table 3. The best read coloring option combination was found when Kermit's color propagation was disabled, which is the reason we did not allow the colors to propagate in the final assembly pipeline.

After inconsistent edges are removed, Kermit starts looking for the unitigs. Instead of finding all maximal non-branching paths, we automatically find all maximal non-branching *rainbow paths*. A rainbow path is like a normal path, but all the colors must appear in increasing order and only once. Kermit loosens this condition by allowing consecutive vertices to use the same color on a path. The removal of all inconsistent edges guarantees that every path we find is also a rainbow path. Kermit outputs these maximal non-branching rainbow path unitigs as our final assembly product.

Abbreviations

GB: Gigabytes; Kbp: Kilobasepairs; MB: Megabytes; OLC: Overlap-layout-consensus; SNV: Single nucleotide variation; VM: Valouev et al. mapper

Acknowledgements

We thank Riku Walve for discussions on guided genome assembly and Christina Boucher for discussions on optical mapping data.

Authors' contributions

LS and ML designed the methods and the experiments. ML implemented the methods and performed the experiments. ML drafted the initial manuscript. ML and LS revised and edited the manuscript. Both authors read and approved the final manuscript.

Funding

This work is supported by the Academy of Finland, via grants 308030, 314170, and 323233 (LS). Academy of Finland had no role in the design of the study, or collection, analysis, and interpretation of data or in writing the manuscript.

Availability of data and materials

OpticalKermit is freely available at https://github.com/Denopia/kermit-optical-maps.
S. cerevisiae reference genome is available at https://www.ncbi.nlm.nih.gov/genome/15?genome_assembly_id=22535.
C. elegans reference genome is available at https://www.ncbi.nlm.nih.gov/genome/41?genome_assembly_id=43998.
A. thaliana reference genome is available at https://www.ncbi.nlm.nih.gov/genome/4?genome_assembly_id=454618.
A. thaliana reads are available at https://downloads.pacbcloud.com/public/SequelData/ArabidopsisDemoData.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 13 December 2019 Accepted: 22 June 2020
Published online: 06 July 2020

References

1. Sedlazeck FJ, Lee H, Darby CA, Schatz MC. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. Nat Rev Genet. 2018;19:329–46.

2. Dimalanta ET, Lim A, Runnheim R, Lamers C, Churas C, Forrest DK, de Pablo JJ, Graham MD, Coppersmith SN, Goldstein S, et al. A microfluidic system for large DNA molecule arrays. Anal Chem. 2004;76(18):5293–301.
3. Samad A, Huff EF, Cai W, Schwartz DC. Optical mapping: A novel, single-molecule approach to genomic analysis. Genome Res. 1995;5(1):1–4.
4. Lin HC, Goldstein S, Mendelowitz L, Zhou S, Wetzel J, Schwartz DC, Pop M. AGORA: Assembly guided by optical restriction alignment. BMC Bioinformatics. 2012;13:189.
5. Alipanahi B, Salmela L, Puglisi SJ, Muggli M, Boucher C. Disentangled long-read de Bruijn graphs via optical maps. In: Schwartz R, Reinert K, editors. 17th International Workshop on Algorithms in Bioinformatics, WABI 2017. Leibniz International Proceedings in Informatics. Germany: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik; 2017. p. 1–14.
6. Walve R, Rastas P, Salmela L. Kermit: Guided long read assembly using coloured overlap graphs. In: Parida L, Ukkonen E, editors. 18th International Workshop on Algorithms in Bioinformatics (WABI 2018). Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik; 2018. p. 1–11.
7. Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Res. 2017;27:722–36.
8. Valouev A, Li L, Liu Y-C, Schwartz DC, Yang Y, Zhang Y, Waterman MS. Alignment of optical maps. J Comput Biol. 2006;13(2):442–62.
9. Nagarajan N, Read TD, Pop M. Scaffolding and validation of bacterial genome assemblies using optical restriction maps. Bioinformatics. 2008;24(10):1229–35.
10. Leung AK-Y, Kwok T-P, Wan R, Xiao M, Kwok P-Y, Yip KY, Chan T-F. OMBlast: alignment tool for optical mapping using a seed-and-extend approach. Bioinformatics. 2017;33(3):311–9.
11. Mendelowitz LM, Schwartz DC, Pop M. Maligner: a fast ordered restriction map aligner. Bioinformatics. 2016;32(7):1016–22.
12. Muggli MD, Puglisi SJ, Boucher C. Efficient indexed alignment of contigs to optical maps. In: Algorithms in Bioinformatics: 14th International Workshop, WABI 2014, Wroclaw, Poland, 8-10 September 2014. Proceedings. Berlin, Heidelberg: Springer; 2014. p. 68–81.
13. Muggli MD, Puglisi SJ, Boucher C. A succinct solution to Rmap alignment. In: Parida L, Ukkonen E, editors. 18th International Workshop on Algorithms in Bioinformatics (WABI 2018). Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik; 2018. p. 1–16.
14. Muggli MD, Puglisi SJ, Ronen R, Boucher C. Misassembly detection using paired-end sequence reads and optical mapping data. Bioinformatics. 2015;31(12):80–8.
15. Pan W, Jiang T, Lonardi S. OMGS: Optical map-based genome scaffolding. J Comput Biol. 2020;27(4):519–33.
16. Dong Y, Xie M, Jiang Y, Xiao N, Du X, Zhang W, Tosser-Klopp G, Wang J, Yang S, Liang J, et al. Sequencing and automated whole-genome optical mapping of the genome of a domestic goat (Capra hircus). Nature Biotech. 2013;31(2):135–41.
17. Ganapathy G, Howard JT, Ward JM, Li J, Li B, Li Y, Xiong Y, Zhang Y, Zhou S, Schwartz DC, et al. High-coverage sequencing and annotated assemblies of the budgerigar genome. GigaScience. 2014;3(1). Article Id 2047-217X-3-11.
18. Vij S, Kuhl H, Kuznetsova IS, Komissarov A, Yurchenko AA, Van Heusden P, Singh S, Thevasagayam NM, Prakki SRS, Purushothaman K, et al. Chromosomal-level assembly of the Asian seabass genome using long sequence reads and multi-layered scaffolding. PLoS Genet. 2016;12(4):1005954.
19. Beier S, Himmelbach A, Colmsee C, Zhang X-Q, Barrero RA, Zhang Q, Li L, Bayer M, Bolser D, Taudien S, et al. Construction of a map-based reference genome sequence for barley, Hordeum vulgare L. Sci Data. 2017;4:1–24.
20. Daccord N, Celton J-M, Linsmith G, Becker C, Choisne N, Schijlen E, Van de Geest H, Bianco L, Micheletti D, Velasco R, et al. High-quality de novo assembly of the apple genome and methylome dynamics of early fruit development. Nat Genet. 2017;49:1099–106.
21. Jarvis DE, Ho YS, Lightfoot DJ, Schmöckel SM, Li B, Borm TJA, Ohyanagi H, Mineta K, Michell CT, Saber N, et al. The genome of Chenopodium quinoa. Nature. 2017;542:307–12.
22. Morisse P, Marchet C, Limasset A, Lecroq T, Lefebvre A. CONSENT: Scalable long read self-correction and assembly polishing with multiple sequence alignment. BioRxiv. 2020;546630.
23. Stöcker BK, Köster J, Rahmann S. SimLoRD: simulation of long read data. Bioinformatics. 2016;32(17):2704–6.
24. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34(18):3094–100.
25. Gurevich A, Saveliev V, Vyahhi N, Tesler G. QUAST: quality assessment tool for genome assemblies. Bioinformatics. 2013;29(8):1072–5.
26. Li H. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics. 2016;32(14):2103–10.
27. Shelton JM, Coleman MC, Herndon N, Lu N, Lam ET, Anantharaman T, Sheth P, Brown SJ, et al. Tools and pipelines for BioNano data: molecule assembly pipeline and FASTA super scaffolding tool. BMC Genomics. 2015;16:734.
28. Waterman MS, Smith TF, Katcher HL. Algorithms for restriction map comparisons. Nucleic Acids Res. 1984;12(1Part1):237–42.

Publisher's Note